Student Name: Edwin Raj S

Seat No: 232

Project ID: 32

Project title: IIC Data Management

Technical Stack:

Frontend	HTML, CSS, Javascript
Backend	PHP Laravel
Database	MySQL
API	RESTful API

1. Introduction

1.1. Purpose:

This document is to describe the process and the workflow of the IIC Data Management (IIC Portal). It will explain the systematic flow and features of the system.

1.2. Problem Statement

The nonsystematic and randomness of the data make the task of uploading the report file to the IIC Portal more time consuming and difficult with problems like:

1. **Inconsistent Formatting:** When data is not structured consistently throughout the report, it becomes challenging to extract and organize the necessary information for uploading.

2. **Data Errors:** Errors in the data, such as typos, inaccuracies, or inconsistencies, can complicate the uploading process and may require manual correction before submission.
3. **Duplicate Entries:** Duplicate entries can cause confusion and errors during the upload process, requiring extra time to identify and resolve these issues.
4. **File Size Limitations:** Large file sizes can make it difficult to upload the report to the IIC Portal, especially if there are restrictions on the maximum file size allowed.

And many more.

1.3. Scope of the Project:

- This system will serve as a portal which enables the student to upload the stage 1 data for the Self-driven activity and wait for the approval, on successful verification the student can conduct the specific activity on the allotted venue and can upload the stage 2 data for the activity within the deadline and claim their rewards.
- The Administrator can have detailed and organized information on the Self-driven activity which is conducted by the students group. Other than that, the administrator can upload the details on Calendar activity, Celebration activity, MIC-driven activity and also on EDC and Startups.

2. System Overview:

2.1. Users:

1. Students (user):

The users are enabled to only view the Celebration, Calendar, MIC driven, EDC and startups activities conducted by the organization.

The users are only enabled to add the Self-driven activity on their own ideas and conduct the activity and upload the proof for the verification within the deadline

2. Staffs (admins):

The users are given with full control over the portal they are:

- Enabled to add Celebration, Calendar, MIC-driven activities.
- Enabled to add EDC and Startups events.
- Enabled to approve or reject the activities proposed by the student organizations at any time.

2.2. Features:

1. Authentication:

With Google OAuth 2.0 integration, login is streamlined for users within your organization (e.g., @bitsathy.ac.in) while unauthorized accounts are restricted.

2. Creating Self-driven activity:

Students can propose their own initiatives for entrepreneurship or innovation through the IIC Portal. Upon approval, completing the activity and submitting proof unlocks their rewards.

3. Activity Status:

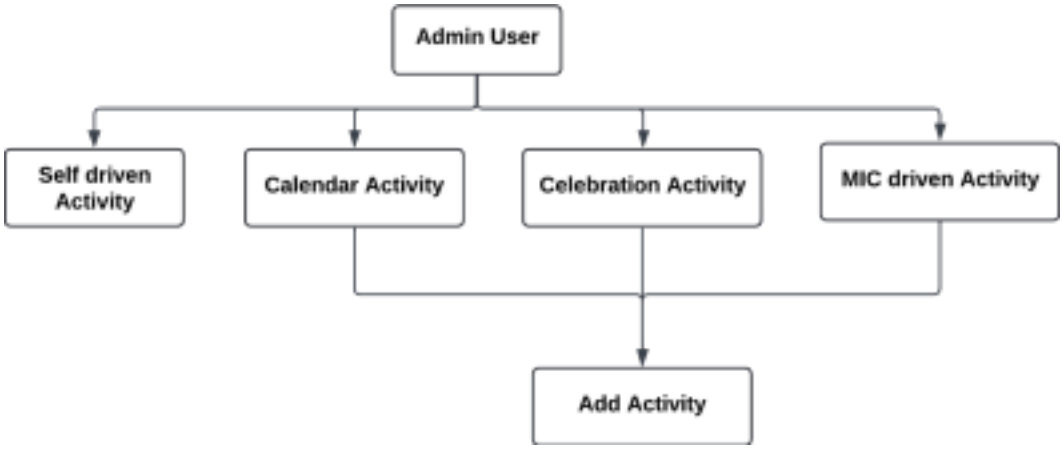
Students can view their activity status in the event logger page under the self-driven activity.

4. Admin Access:

Admin can approve or reject the activity at any time and also add new activity on the IIC Portal.

Student Interaction:

Admin Interaction:



3. Flow chart:

3.1. Functional Requirements:

1. User Management:

The IIC Portal provides user management such as:

- The system must allow students to register and create accounts using their institutional email addresses.
- The system should have functionalities for student profile management

2. Activity Proposal System:

The IIC Portal must have the followings:

- The system should allow students to submit proposals for their self-driven activities.
- Proposals should include details like activity description, objective, time and outcome.

3.Approval Workflow:

The IIC Portal should have a designated team or role responsible for reviewing student proposals. The system should facilitate the review process with functionalities like status tracking, reviewer comments, and approval/rejection decisions.

4.Proof Submission:

The following are the requirements for proof submission:

- Once a proposal is approved, students should be able to submit proof of completing their activity.
- The system should allow uploading documents, photos, or other evidence formats.
- It should allow the students to edit the logger which has been rejected.

3.2. Non-Functional Requirements:

1. Performance:

The system should be able to handle a high volume of student users and activity proposals without significant delays. Page load times and response times for user interactions should be within acceptable limits.

2. Security:

- The system should employ secure user authentication methods.
- Data security is crucial, ensuring student information, activity proposals, and proof of completion are protected from unauthorized access.

3. Scalability:

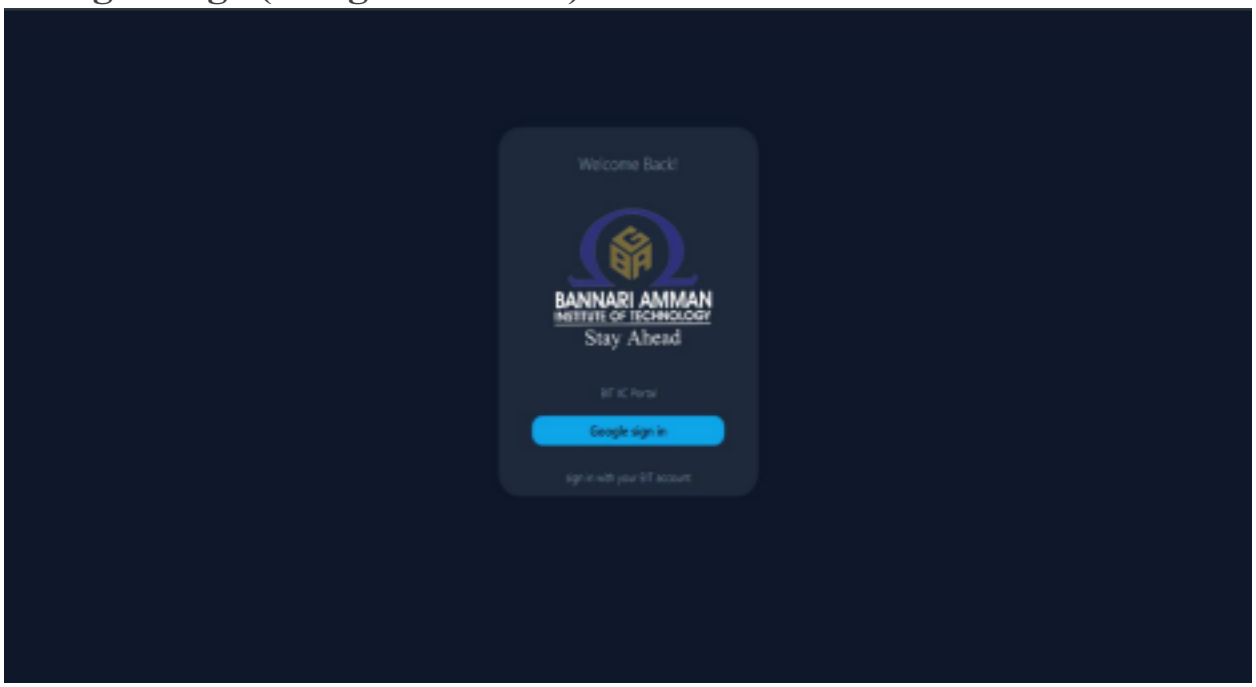
The system should be designed to accommodate future growth in the number of users, activities, and reward programs. This might involve using scalable architecture or infrastructure.

4. Error Handling:

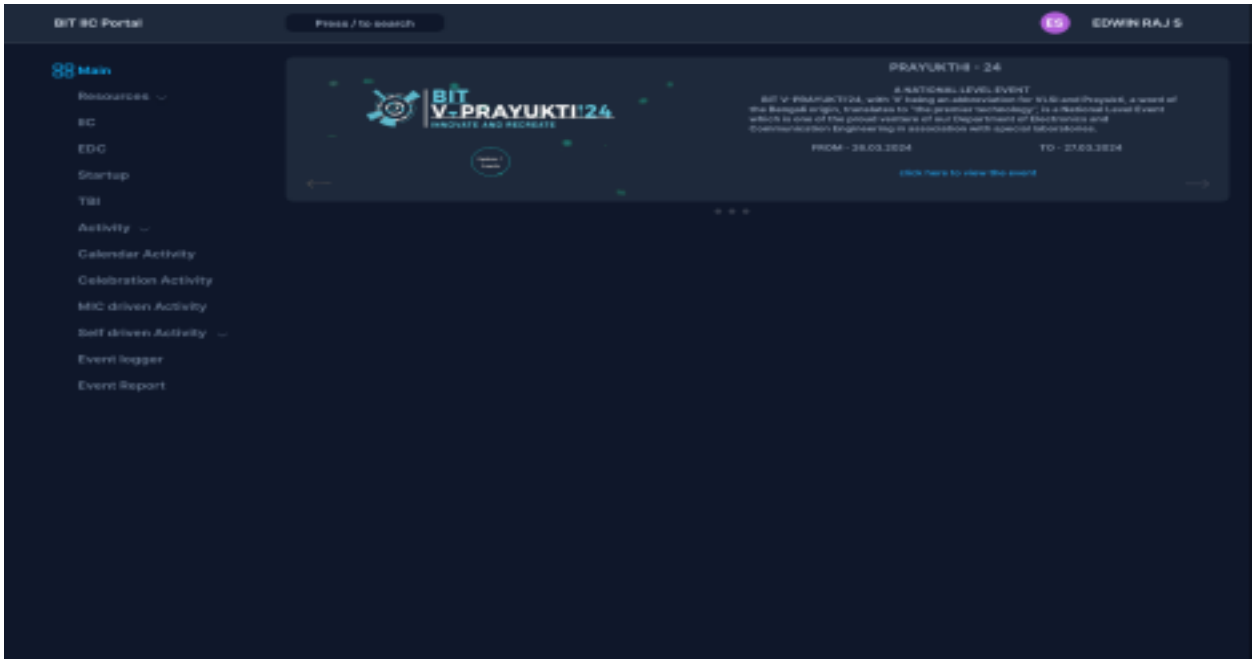
The system should handle errors gracefully, providing clear and informative messages to users encountering issues during proposal submissions, proof uploads, etc. Robust error logging can be beneficial for troubleshooting and identifying potential problems.

4. Prototype of the Project:

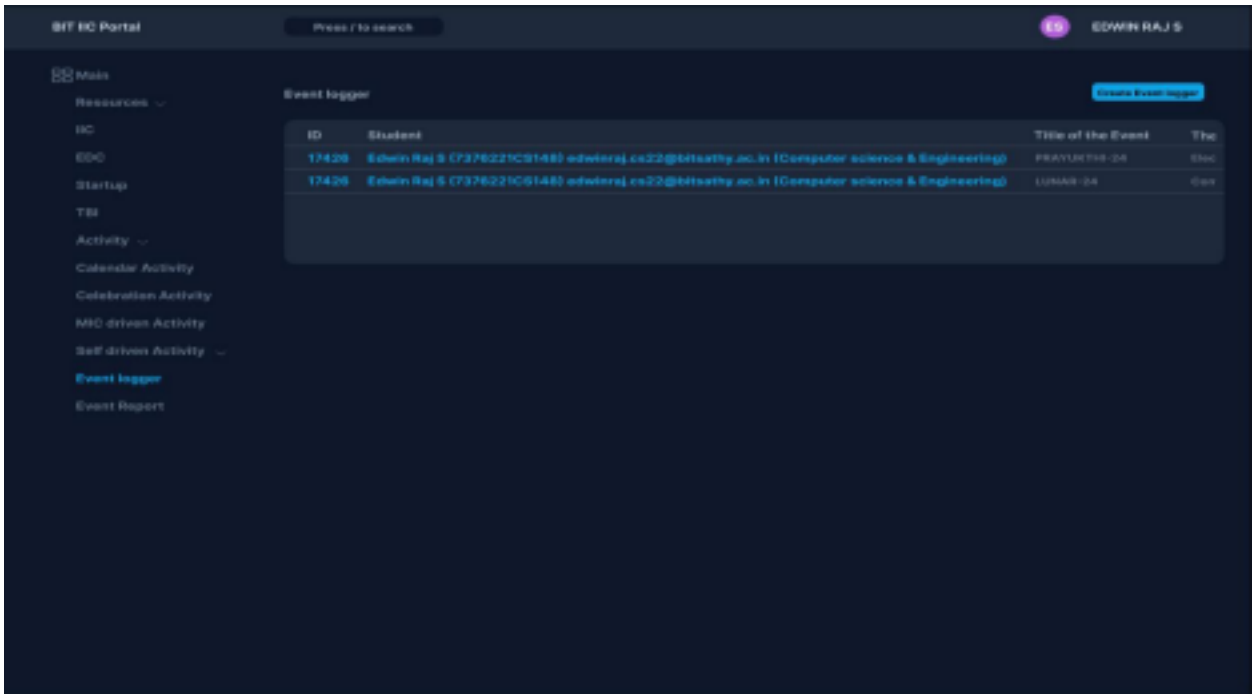
1. Login Page (using OAuth 2.0):



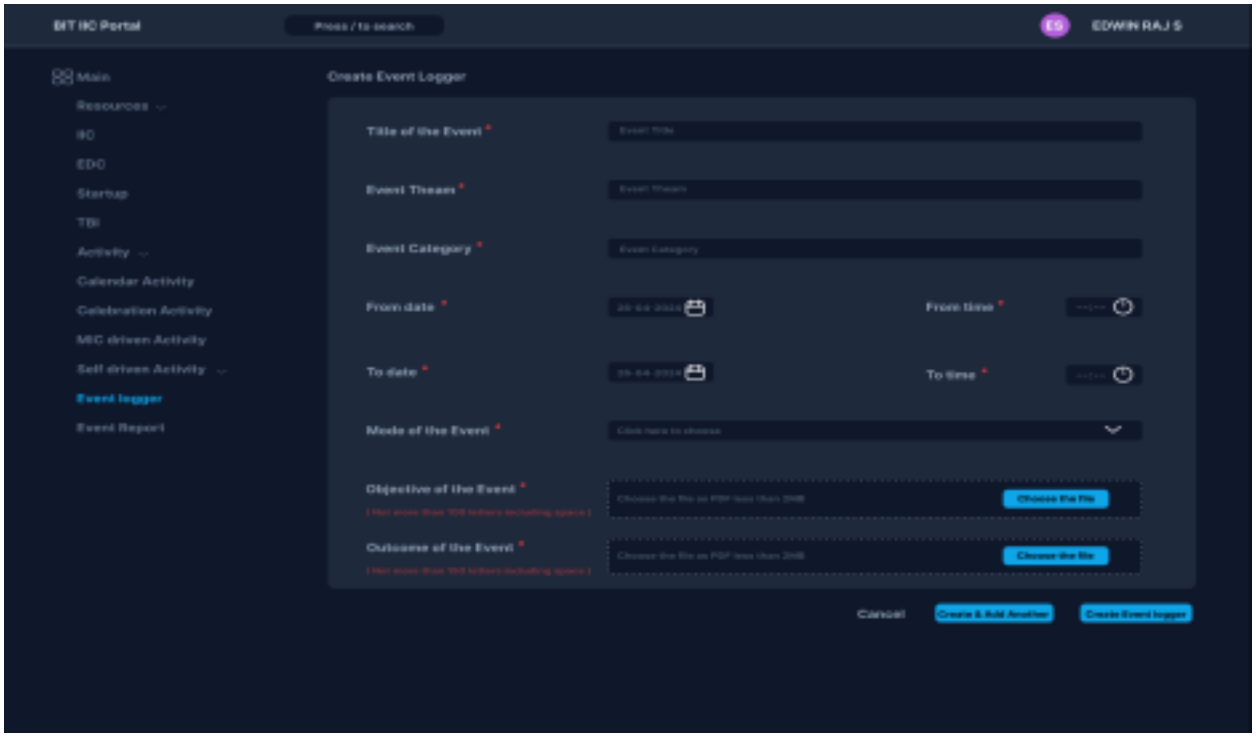
2. Landing Page:



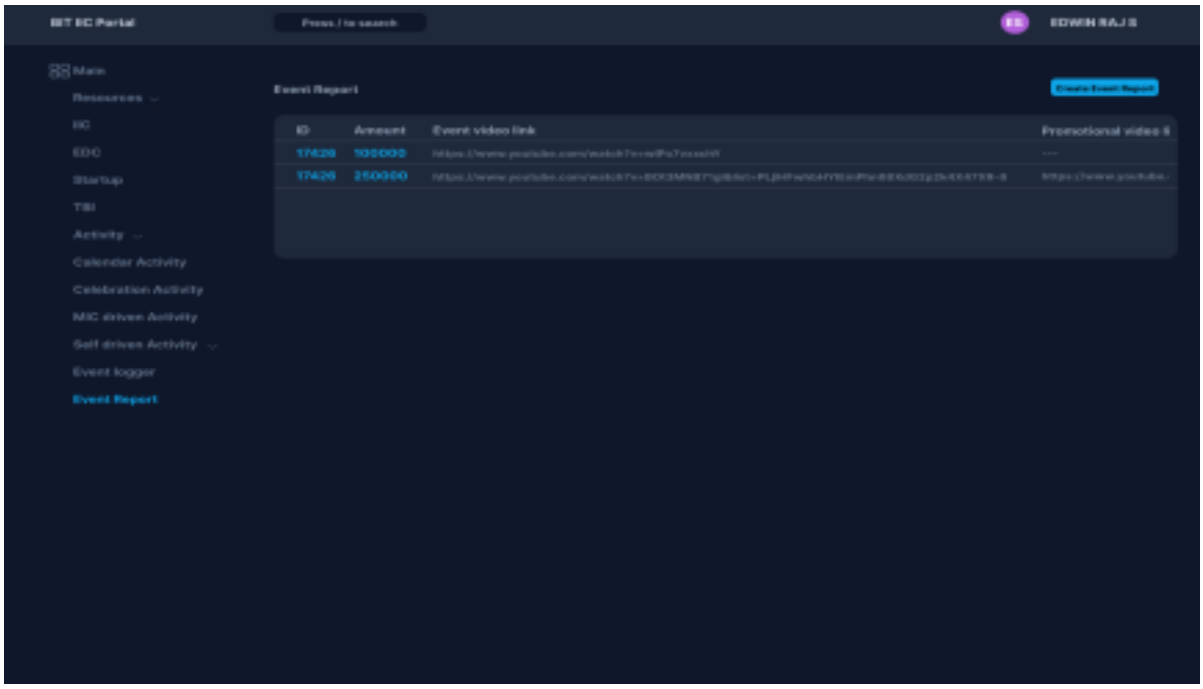
3. Adding Self-driven activity:



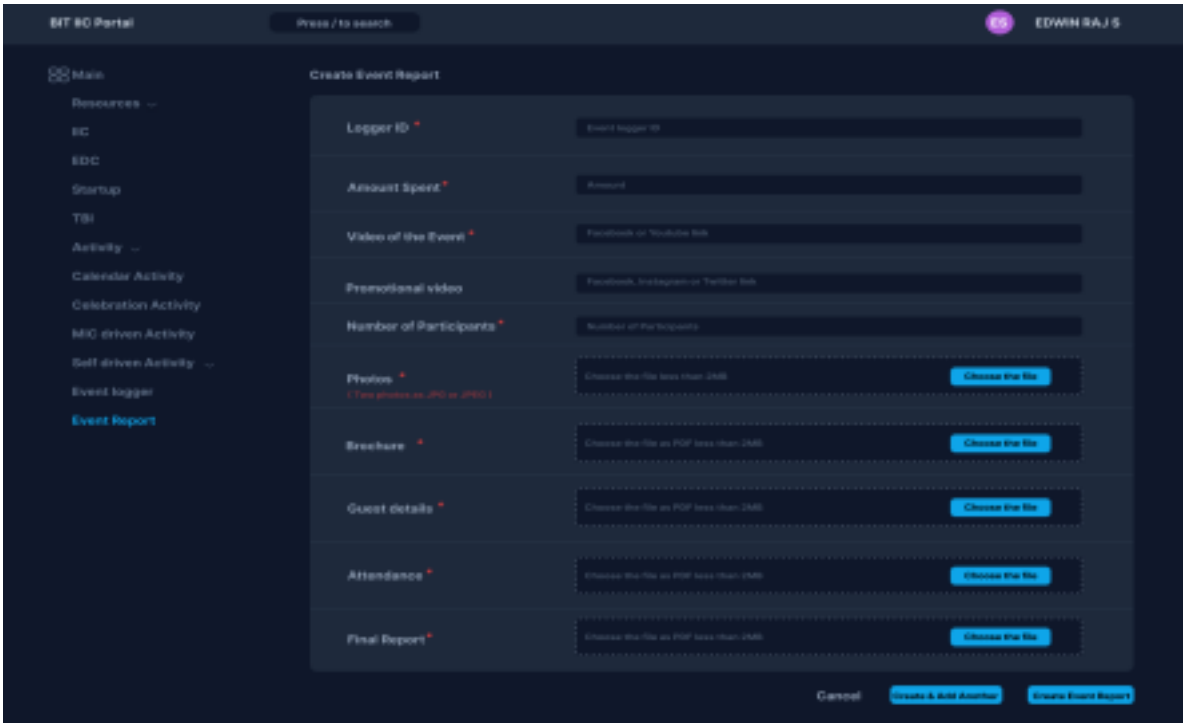
4. Creating Self-driven activity:



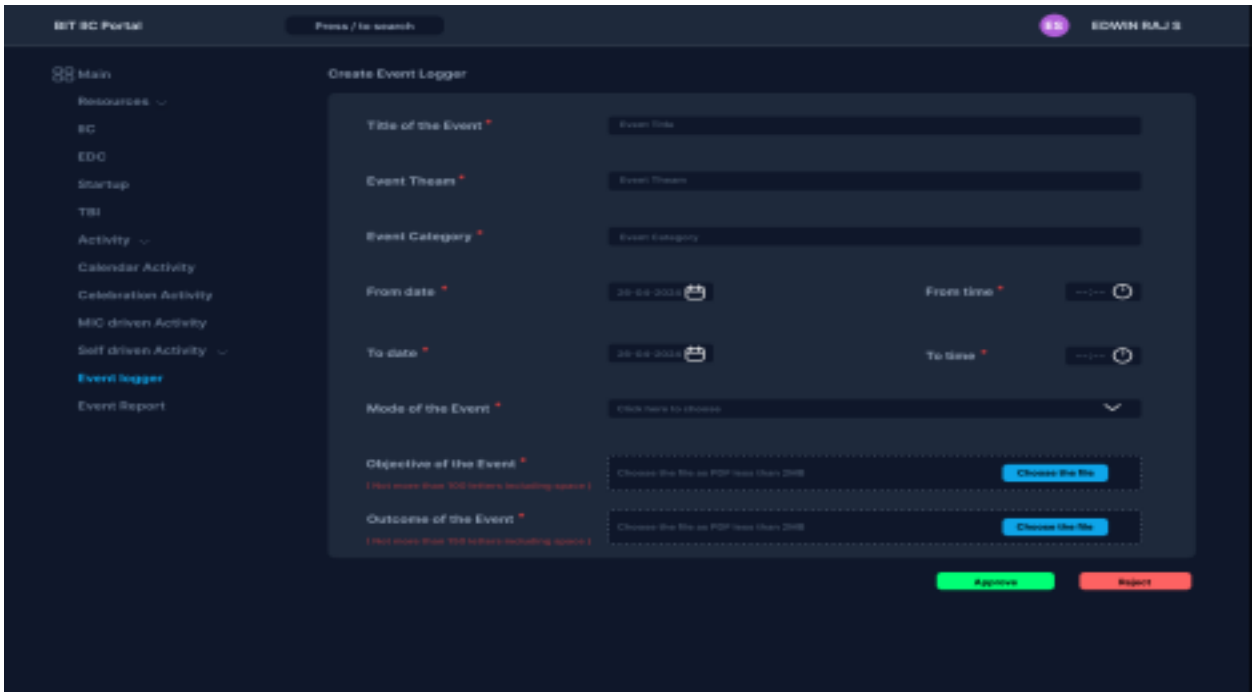
5. Adding Self-driven activity report:



6. Creating Self-driven activity report:



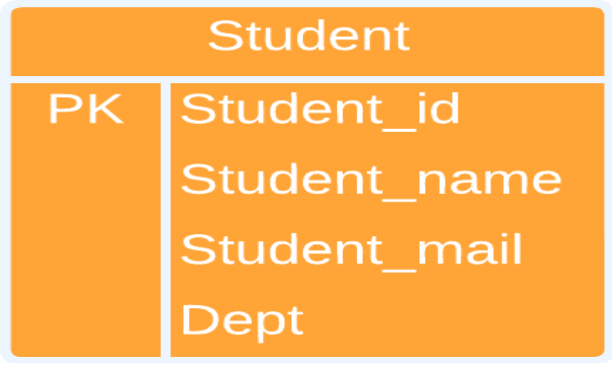
7. Admin page for self-driven activity:



5. Database Management:

5.1. Data schema:

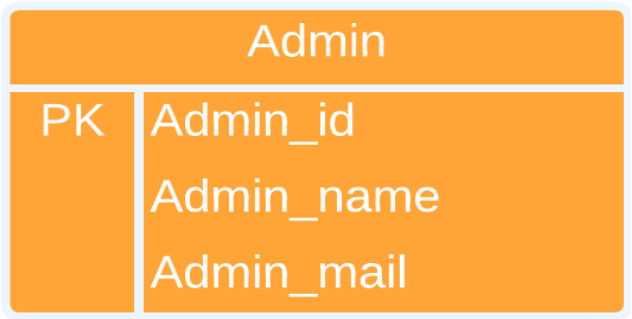
1. Student Entity:

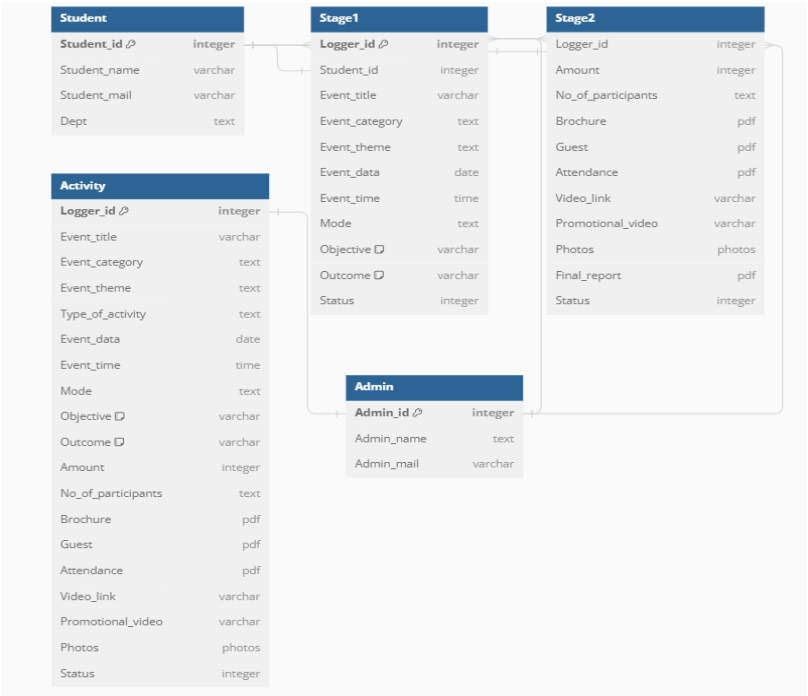


2. Stage-1 Entity:

3. Stage-2 Entity:

4. Admin Entity:



5. Activity:

Entity Relationship Diagram (ERD)